A Presentation of Structures and Applications of Convolutional Neural Networks

Minghao Bai [1], Muxian Li [2, *]

[1] Cambridge International School, Daqing No.1 Middle School, Daqing 163000, China

[2] Nanjing Foreign Language School, Nanjing 210000, China

* Corresponding author's e-mail: muxianli@hhu.edu.cn

Abstract. This paper reviewed the history of convolutional neural networks, why and how they developed, and what inspired the scientists to design them. To make CNNs simpler to understand, we wrote about their characteristics and structures while introducing the basic units of convolutional neural networks, including training and modeling parameters and how they would affect the confidence and efficiency of the whole process, different kinds of layers and how they work, multiple pooling methods and loss functions with their formulas. This paper also included applications of convolutional neural networks in computer vision and natural language processing while specifying and analyzing the technologies in use to clarify this introduction. Challenges and future research directions of the convolutional neural networks were pointed out to help refine this technique.

Keywords: Structure, Application, Convolutional Neural Network.

1. Introduction

With the changes of the times, science and technology are also advancing with the times. As the era of big data has progressed, more complex convolutional neural networks with many layers have proven to be better at learning and expressing patterns than traditional machine-learning algorithms. Following their development, convolutional neural networks and other deep learning models have achieved new levels of accuracy in many large-scale computer vision applications that recognize objects. Convolutional neural networks have been utilized in a wide range of contexts, where they've made significant advances. This passage discusses the history of deep learning and convolutional neural networks, alongside their basic structures, and how they extract features from images. It also talks about where these models are being used today in computers.

2. Characteristics and structure of CNNs

2.1. Characteristics

Artificial neural networks are computing systems that imitate the way biological brains work. They consist of large groups of interconnected processing units, called neurons, which learn by continuously adjusting their internal weightings to improve their performance at outputting accurate responses whenever they receive a specific input from another unit. Artificial neural networks that employ convolutional layers are the best-known type of such networks.

Primarily used to analyze visual or sound input, they now have applications as diverse as image recognition, image segmentation, and natural language processing [1].

Since primary neural networks are interpreted with multilayer perceptrons, in which every node connects to each other, it does not have a mechanic to prevent overfitting by themselves; Instead, they have to use methods like weight decaying or dropping out random neurons. Convolution kernels eliminate this drawback, as they extract subtle patterns in the input dataset to increase complexity and uniqueness.

2.2. Structure

In a convolutional network, convolutional, pooling, fully connected and loss layers are the four types of layers.

2.2.1. Convolutional layers

Convolutional layers are the fundamental elements in a convolutional network, through which the input tensor with a dedicated shape is processed by a convolutional filter, including a series of kernels, sliding across the input tensor [2].

The output tensor will have its height and width reduced by the height or width of the kernel minus one while keeping other dimensions the same. The principle of the sliding kernel is that it starts calculating from the top left of the input tensor, giving a one-dimensional sum of product on each calculation set, then moving on to the next area until it reaches the end. A figure of how convolutional cores work is shown in figure 1.

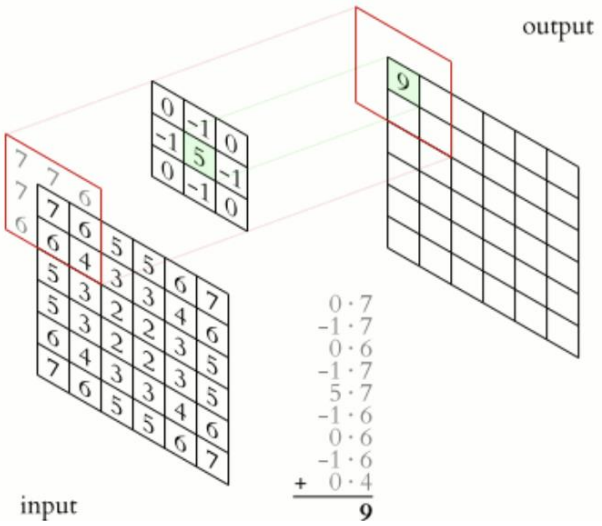

Figure 1. A figure of how convolutional cores work

2.2.2. Pooling layers

Pooling layers appear after convolutional layers to generate orderly data capable of being processed by layers following. It reduces the amount of data that needs to be processed by using the outputs from the former layer as inputs to a single neuron in the latter. Rather than being a trained algorithm, it's a specified process.

Pooling layers produce a lower-resolution version of the input by combining and averaging groups of adjacent pixels. This process preserves important local features, regardless of whether those features are detected at exactly the same location on each image within the given dataset.

There are several kinds of pooling layers.

Maximum pooling. Maximum pooling is a simple method of separating graphs into blocks, while saving the most significant values of the blocks as the pooling results. While adjusting parameters reversely, the pooling results are transmitted to smaller blocks with previously recorded IDs, $maxID_i$.

Average pooling. Average pooling is a simple method of separating graphs into halves then using the average value of each block as the pooling result. While transmitting backwards, the pooling result is evenly separated into halves then given to smaller blocks from the earlier layer.

Lp pooling. Lp pooling is derived from the way biological cells work. This method is commonly seen in CNN for it saves background and texture information of the graphs by deleting some other characteristics as a trade-off.

Random pooling. Random pooling is designed based on Dropout [3]. This method would randomly choose a value in its pool via a specific probability distribution to ensure that some of the smaller stimulation signals will reach the next layer.

Mixed pooling. Mixed pooling is designed based on Dropout and DropConnect[4]. This method can be recognized as a linear composition of random pooling and maximum pooling.

Spatial pyramid pooling. Unlike fully connected layers which require a set value of dimension to process the input matrix correctly, spatial pyramid pooling is designed to transform an image of any dimension into a same dimensional output. This kind of pooling can not only process more than one kind of photos, but also can avoid cropping and wrapping, which would cause losses of information.

Spectral pooling. Spectral pooling is a method based on fast Fourier transform, which can build an FFT-based CNN with a FFT convolution layer. During the calculation, this method will DFT transform every channel of the input graph, where a $N * N$ matrix is extracted to be the pooling result. This kind of pooling also comes with filtering functions to save low frequency characteristics while adjusting the sizes of the graphs.

2.2.3. Fully-connected layers

Similar to common ones, every neuron in a fully connected layer connects to all the neurons being properties of the previous and the following layer. The final classification is also done in this layer. Their activations are computed by applying affine transformations to the original inputs and adding bias terms.

2.2.4. Loss layers

In order to prevent generating false outputs, loss layers are added to the models. The loss parameter specifies how the networks are penalized when they produce outputs that differ from what were expected. There are various loss functions, depending on the specific task trying to be accomplished. Different loss functions can be used to optimize different systems. Below are some of the most commonly used ones.

Classification error. The most direct definition of this function is in formula 1.

$$ClassificationErr = \frac{Count\ of\ error\ items}{Count\ of\ all\ items} \tag{1}$$

However, this model can only judge which network made less mistakes, instead of calculating how wrong it was. The definitions of Mean squared error and mean absolute error are shown in formula 2 and 3.

$$MSE = \frac{1}{n}\sum_{i}^{n}(\hat{y_i} - y_i)^2 \tag{2}$$

$$MAE = \frac{1}{n}\sum_{i}^{n}|\hat{y_i} - y_i| \tag{3}$$

These two models work well while calculating the accuracy of the model. However, these models are not commonly used since they are slow and low-efficient while cooperating with the Gradient Descent Method.

SoftMax loss. To classify a specific variable into several independent categories, the SoftMax loss function can be utilized.

The formula for calculating SoftMax is shown in formula 4.

$$S_i = \frac{e^{zi}}{\sum_{j=1}^{K} e^{zj}} \tag{4}$$

Where z represents the input matrix, Z_i and Z_j represent the i^{th} or the j^{th} element of the matrix. Due to the exponential operation, the element with the largest confidence value, will dominate the result.

However, when SoftMax is used as a layer, the produced data should be sifted through as it possesses a trend of interpreting values close to 0 or 1.

Based on this formula, the formula for SoftMax loss looks like in formula 5.

$$L = -\sum_{j=1}^{K} y_j * log\ (S_j) \tag{5}$$

Where y_j represents a $1 * K$ matrix whose values equal 0 except for the j^{th} one equals 1. This characteristic allows the formula to be simplified as formula 6 follows:

$$L = -log\ (S_j) \tag{6}$$

Which means that the loss of a wrong prediction is more significant than a correct one, while the loss of a terrible prediction is more significant than a relatively accurate one.

However, in 2019, Deng and other colleagues described two disadvantages of SoftMax loss: large parameter sets and separated characteristics for closed datasets. Moreover, applications like face recognition introduce a more than larger open datasets, which will have more data constantly flowing. In this case, SoftMax loss function will be slowed down on a large scale [5].

Cross-Entropy loss. The Cross-Entropy loss function is used to predict probabilities from 0 to 1, where the probabilities are not necessarily dependent of one another.

The formula for it is shown in formula 7.

$$L = \frac{1}{n}\sum_i L_i = -\frac{1}{n}\sum_i \sum_{j=1}^{K} y_{ij} * log\ (p_{ij}) \tag{7}$$

Where i ranges from 0 to $n - 1$ to represent each of the elements and j represents one element that is being calculated for L_j.

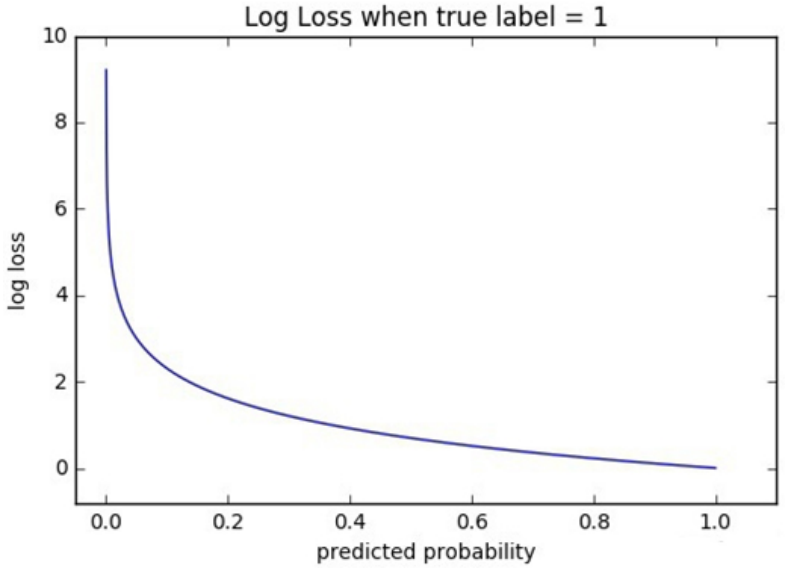

Figure 2. A graph of log loss – predicted probability when valid label equals 1

Figure 2 plotted for this function shows that the best loss value – predicted probability pair can be calculated through first differentiation.

Euclidean loss. Euclidean loss is used for regression problems where the training datasets include real-valued labels, mostly in time-series models. This model would try to compose multiple curves together.

However, in real-life cases, since observed sequences may fluctuate due to noises, a Euclidean loss function calculated result may contain multiple randomly located spikes.

3. The practices of convolutional neural networks

3.1. Implementations of convolutional neural networks in computer vision

With the advent of big data, improved computing power and new algorithms, deep convolutional neural networks, which are trained by deep learning algorithms, are now able to do things that were previously impossible or impractical, and have proven remarkably effective at solving many large-scale recognition tasks in computer vision [6]. From a Bioscience perspective of view, computer vision attempts to initiate arithmetic models from humanly optic means. In terms of Engineering, computer vision refers to autonomous systems based on the abilities of human visual nerves. The research and development of convolutional neural network models are helping to advance computer vision capabilities in many application fields, such as image classification, object detection, pose estimation, and face recognition. They are mainly presented in three aspects: the construction of typical networks, training methods used to create them and their performance as measured by key metrics.

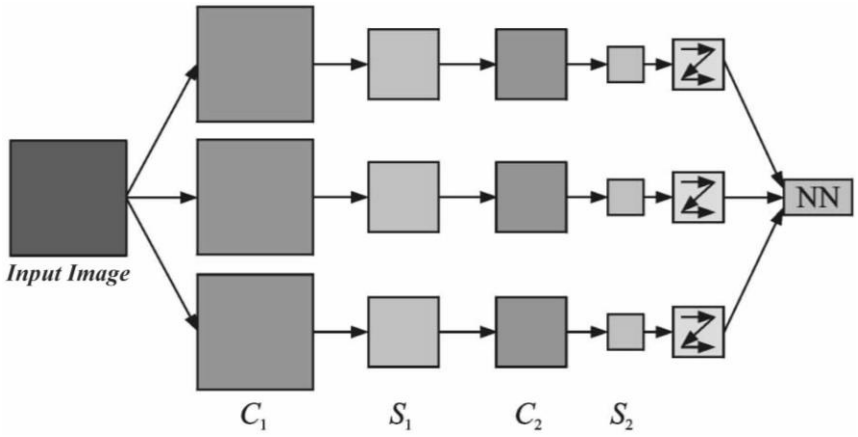

Figure 3. A graph of a diagrammatic illustration of a CNN

The figure 3 is a diagrammatic illustration of a convolutional neural network. The network model consists of two alternating convolutions $(C1, C2)$ and two subsampling layers $(S1, S2)$. To generate the three feature maps, each original image is convoluted with three trainable filters and biasing vectors. Then in each local area of a feature map, all pixels, whose distance from (or proximity to) that pixel falls within a given range, were averaged together. After adding bias terms and applying nonlinear activation functions, three new feature maps are generated at layer $S1$. After this, these feature maps are applied with three trainable filters in $C2$, and output three more refined feature maps after passing through $S2$. Finally, the three outputs are vectorised respectively and used to train a conventional neural network.

3.1.1. Object detection

Object detection technique can help with recognizing, detecting and localizing multiple visual items in images or continuously in successive video frames. Rather than basic object classification, this method better understands detected objects as unique individuals. With this method, real-time locations and numbers of certain objects are labelled beside them.

In this area, a more complicated version of object detection is the defect classification method. It can also follow inspection processes. Based on different sample methods, this method can help determine how many items should be inspected and how many defects in a certain batch can trigger a failed inspection. Since it is difficult to separate the background and defects, the characteristics of defects can be combined with the visual saliency models and double low-rank representation models to realize the separation processes. Since the number of layers, the computational scale, and the complexity increase simultaneously, generally, uncompressed CNNs cannot be applied to embedded platforms, which partially meets the industrial real-time detection, since it needs some hardware support. However, newer models can be converted to simplified forms where all the terms and

weights are integers to accelerate the calculation on SOCs, which don't usually come with high-speed float point units.

3.1.2. Image classification

CNN, on the basis of deep learning, can be used for image identification and classification. This method automatically extracts important features from large amounts of data, enabling computers to understand images better. At present, most methods of image classification directly use common convolutional networks to classify images, such as AlexNet [7], VGG [8], or GoogleNet [9]. They have proved their application values in a competition to identify objects in digital images known as the ImageNet Large Scale Visual Recognition Challenge.

3.2. Application of convolutional neural networks in natural language processing

Natural language processing (NLP), also known as natural language understanding (NLU), is an essential branch of artificial intelligence research. It mainly studies how computers can process various types of human languages to read like human beings. Research contents in natural language processing are pervasive, mainly including text classification, emotion classification, machine translation, intelligent question answering, and speech recognition. In these aspects, research results have been applied to areas including search engines, recommendation systems, smart robots, intelligent customer service, and voice input methods.

3.2.1. Text classification

Text classification is one of the most widely used aspects of neural networks. It automatically classifies the text into categories by automatically recognizing the features of the texts or expressed emotions for subsequent processing. To understand texts better, passages are usually split up into four levels of scopes: Document, paragraph, sentence, and phrase level. Commonly used feature extraction techniques include Term Frequency, Global Vectors for Word Representation and so on. To accelerate the whole process, some of the terms can be conformed into single variables with methods like Principal Component Analysis, Non-negative Matrix Factorization, and Linear Discriminant Analysis.

Text classification processes mostly start with word vector matrixes embedded from original strings, which works greatly as inputs to be convoluted with multiple convolution kernels corresponding to the vector spaces. Then through the pooling methods, down-sampling operations, multiple feature representations of the input can be obtained.

3.2.2. Emotion analysis

For emotion recognition, neural networks can identify humane psychological activities, which plays a decisive role in the further development of intelligent robots. A common method of emotion analysis is to introduce CNNs to classify and discriminate emotions through dynamic modelling and classification of statements to predict and infer possible emotional events. At present, existing researches have achieved relatively excellent evaluation results. In order to determine the emotions of a specific expression, aspect vectors could be embedded while applying convolution to reach a competitive result across all languages and fields.

4. Challenges and prospects

From the current development situation, neural networks such as text classification or emotion recognition has made remarkable achievements, but still, there are many defects to be fixed. The most obvious one is the lack of training sets and computational power, which is the main reason why neural networks are not widely used in real life. Collecting this kind of dataset costs a lot of time and money, especially when optimizing them manually. Also, due to the limitations of the current chip manufacturing process, components can be tough to be much smaller than it is now. The amount limit of electric components limits the power of the hardware [10].

In this case, exploring unsupervised learning of convolutional neural networks is desired to save human power to develop algorithms or structures with higher efficiency.

To make CNNs run faster, adjustments of hyperparameters also help, including reducing convolution kernel sizes, shortening epochs, increasing networks' widths and depths, and generating new pooling methods.

5. Conclusion

This passage has included different kinds of network structures and applications. Multiple pooling and loss functions are also introduced with specific characteristics and formulas. From these historical researches and derived formulas, we hope that in the future, more efficient ways of revising datasets or training models will be found with a smaller loss and higher confidence.

By reading this passage, students or professionals working on this will be provided with a broader idea of how convolutional neural networks can work and how they could make use of it to generate new models to improve the efficiency of human work or, in a more profound way, liberate the productivity.

We hope by the development of convolutional neural networks that face detection could be applied all across the world, that people could have their credentials combined with their identity; that biological metrics could be applied with encryption, and that our data would be better protected; that by generating voice pulse, that we could use it to guide the blind or help the vocally disabled make sounds again.

From researches in the history of CNNs, it is known to all that no new network structures or algorithms are completely split from former versions of them; They are derived from earlier versions of networks and algorithms and they are optimized to work with higher speeds.

Nowadays, technologies iterate in a fast way, it is hard to imagine how CNNs will be like in the next decades or how deep learning developed to support new applications. But what we do know is, though some day in the future, this passage might become outdated without the latest technologies, it can still help professionals design new algorithms or analysis methods as a relatively primitive form. It will still help people understand AI in a simpler and internal way to make them trust and support AI development.

References

[1] Collobert R, and Weston J 2008 A unified architecture for natural language processing *Proceedings of the 25th International Conference on Machine Learning - ICML '08.*

[2] Mostafa S and Wu F X 2021 Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images *Neural Engineering Techniques for Autism Spectrum Disorder* 23 – 38.

[3] Szegedy C, Vanhoucke V, Ioffe S, Shlens J and Wojna Z 2016 Rethinking the Inception Architecture for Computer Vision *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2016 2818 - 26.

[4] Gong Y, Wang L, Guo R and Lazebnik S 2014 Multi-scale Orderless Pooling of Deep Convolutional Activation Features *Computer Vision – ECCV 2014* 392 – 407.

[5] Deng J, Guo J, Yang J, Xue N, Cotsia I and Zafeiriou S P 2021 ArcFace: Additive Angular Margin Loss for Deep Face Recognition *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1.

[6] Lu H and Zhang Q 2016 Applications of deep convolutional neural network in computer vision 1 - 17.

[7] Krizhevsky A, Sutskever I and Hinton G E 2012 ImageNet classification with deep convolutional neural networks *Communications of the ACM* **60** 84 – 90.

[8] Simonyan K and Zisserman A 2014 Very Deep Convolutional Networks for Large-Scale Image Recognition.

[9] Szegedy C, Liu W, Jia Y Q, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 7 – 12.

[10] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J and Chen T 2018 Recent advances in convolutional neural networks *Pattern Recognition* **77** 354 - 77.